

I hereby certify that this paper and/or fee is being deposited with the United States Postal Service 'EXPRESS MAIL POST OFFICE TO ADDRESSEE' service under 37 CFR 1.10 on December 10, 2001 and is addressed to: Assistant Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.

Karen Orzechowski

Signature

DATE OF DEPOSIT: December 10, 2001

EXPRESS MAIL LABEL NO. EL888549318US

Inventor(s): A. M. Rincon, Franc E. Noel, Charles E. Kuhlmann, N. C. Strole, and C. Steven Lingafelt.

**FIELD PROGRAMMABLE NETWORK PROCESSOR AND METHOD
FOR CUSTOMIZING A NETWORK PROCESSOR**

CROSS-RELATED APPLICATIONS

The present application is related to the following listed seven applications:

Serial No. _____ (RPS920010126US1), entitled "Method and System for Use of an Embedded Field Programmable Gate Array Interconnect for Flexible I/O Connectivity;"

Serial No. _____ (RPS920010127US1), entitled "Method and System for Use of a Field Programmable Gate Array (FPGA) Function Within an Application Specific Integrated Circuit (ASIC) to Enable Creation of a Debugger Client Within the ASIC;" Serial No.

_____ (RPS 920010128US1), entitled "Method and System for Use of a Field Programmable Function Within an Application Specific Integrated Circuit (ASIC) To Access Internal Signals for External Observation and Control;" Serial No. _____

(RPS920010129US1), entitled "Method and System for Use of a Field Programmable Interconnect Within an ASIC for Configuring the ASIC;" Serial No. _____

(RPS920010130US1), entitled "Method and System for Use of a Field Programmable Function Within a Chip to Enable Configurable I/O Signal Timing Characteristics;" Serial No. _____ (RPS920010131US1), entitled "Method and System for Use of a Field Programmable Function Within a Standard Cell Chip for Repair of Logic Circuits;" and Serial No. _____ (RPS920010132US1), entitled "Method and System for Use of a Field Programmable Gate Array 9FPGA) Cell for Controlling Access to On-Chip Functions of a System on a Chip (S)C) Integrated Circuit;" assigned to the assignee of the present application, and filed on the same date.

FIELD OF THE INVENTION

The present invention relates generally to network processors and specifically to network processors that are field programmable.

BACKGROUND OF THE INVENTION

Network processors, the engines that control and route Internet traffic must constantly evolve to meet increasing performance demands, provide new functions and support new hardware and software interfaces. The amount of time available to design and produce these processor chips is shrinking in order to meet today's time to market demands. These network processors are designed primarily using one of two methods:

1. A general purpose network processor (NP) chip produced in volume by the NP manufacturer, sold to multiple customers and customized with software to implement customer-specific functions.
2. Application specific integrated circuit (ASIC) chips where customer-specific

functions are implemented in hardware and produced in smaller volumes for a single customer.

There are advantages and disadvantages to both of these approaches. The general purpose network processor is generally more cost-effective than the ASIC solution because a single part number can be produced in large quantities and be purchased for a lower cost than a low-volume custom chip solution and can also provide faster time to market because customer-specific functions are coded in software, not implemented in hardware. The disadvantage is that the customer-specific functions are limited to the performance achievable with software, which may be significantly slower than if the function were implemented in silicon.

The primary advantage of the custom ASIC solution is performance because the hardware is optimized to each customer's function. The obvious drawback is the cost and time-to-market associated with developing customer-specific hardware.

In both the general purpose NP chip and ASIC implementations, the chip designer must decide on exactly what functions and input/output (I/O) interfaces will be supported before committing the design to silicon. Changes required to support changing I/O standards or fix errors found in high-risk logic result in costly hardware respins and potential loss of market.

What is needed is the ability to combine the flexible features on the programmable general purpose processor with the ability to adapt that hardware in the field to customer-specific functions, implement bug fixes and respond to changing I/O interface standards.

The present invention addresses such a need.

SUMMARY OF THE INVENTION

A network processor is disclosed. The network processor comprises a plurality of standard cells; and at least one field programmable gate array (FPGA) cell that can

communicate with at least one of the standard cells. The at least one FPGA cell can provide a specified function based upon field programming techniques to allow for customization of the network processor.

Utilizing a method and system in accordance with the present invention, a network processor can be customized to implement a variety of functions in hardware using embedded FPGA macros. The combined technology of ASIC standard cells plus FPGA cells enables fast time-to-market for new designs while optimizing cost and performance. In addition, the combined ASIC plus FPGA on a single die allows the chip developer to use proven standard cell macros for common logic and programmable cells for high-risk logic. Through a system and method in accordance with the present invention a business process is also provided whereby an ASIC customer can either submit a custom logic file to a vendor or choose from a library of functions to program into the FPGA portion of the chip.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a functional block diagram of a network processor.

Figure 2 is a block diagram of a system for customizing a field programmable network processor (FNP).

Figure 3 is a flow chart, which illustrates a method for customizing a field programmable network processor in accordance with the present invention.

DETAILED DESCRIPTION

The present invention relates generally to network processors and specifically to network processors that are field programmable. The following description is presented to

enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

A field programmable network processor in accordance with the present invention combines the flexibility and time-to-market of a general purpose network process with the ability to customize portions of the silicon in the field. This can be accomplished by adding field-programmable gate array (FPGA) cells to the general purpose network processor chip. Customer-specific functions, high-risk logic and I/O interfaces can be implemented in the FPGA portion of the chip in the field using standard FPGA programming techniques. FPGA cells can be manufactured on the same silicon die as standard cell gates using existing manufacturing techniques.

Figure 1 is a functional block diagram of a network processor 100. Although a particular type of network processor is shown, one of ordinary skill in the art recognizes that the present invention could be utilized with a variety of network processors that include many of the same elements or different elements and that use would be within the spirit and scope of the present invention.

As is seen, the network processor 100 includes a processor local bus (PLB) 102, and two on-chip peripheral busses (OPB) 104 and 106. As is also seen, there are a plurality of functional blocks coupled to these busses 102, 104 and 106. As is seen, a rapid I/O 108, an accelerator function 110, a PLB master/slave function 112, SDRAM controllers 114 and 116

and an OPB bridge 118 are all in communication with the PLB 102.

The OPB bridge 118 is also in communication with an on-chip peripheral bus 106, which also includes an arbitration unit 120. There are a plurality of devices that interact with the on-chip peripheral bus 106. As is seen, they include a peripheral controller 122, a UART 124, a GPIO preprocessor 126, a GPIO 128, an OPB master/slave function 132. An I²c device is also coupled to OPB 106. There are a plurality of devices, as is seen, that communicate from the PLB 102 to the OPB 104. They include the look up classifier 150, a buffer management header cache 152, a frame alteration function 156 and egress queues 158. A master controller 154 is also coupled to the processor local bus.

The PLB 102 also communicates with a microprocessor 160, a media access layer (MAL) function 164 which is in communication with a media access controller 134. An interrupt controller 166 receives signals from the outside, which are provided to the microprocessor 160. The microprocessor 160 includes a first and second instruction caches, a memory management unit (MMU), a JTAG function and a trace cache. There are also a plurality of media interfaces 170 coupled to the OPB 104.

There are several elements, which could be implemented as field programmable functions. These are primarily functions that are functions that could be changeable depending upon a particular customer involved. Accordingly, it is desirable that several of these blocks be changed into FPGA cells to allow them to be programmable in the field. In this example, it has been determined by experience and empirical evidence that the accelerator function 110, the PLB master/slave functions 112, the GPIO preprocessor 126, the on-chip media interfaces 170, and the OPB master/slave function 134 could all be implemented as FPGA cells to customize the network processor.

Below is a representative list of functionality within the functional blocks, which could be customized.

1. Hardware Accelerator Function 110

- Bandwidth Allocation Technology (BAT)
- IP Header Compression
- Checksum Computation
- Protocol Conversion
- Hardware Conversion function for gateway
- Layer Two bridge function

2. PLB Master/Slave Function 112

- Customer Proprietary Logic
- PowerPC Performance Monitor
- PCI/PCI-X
- Chip-to-Chip Interface Macro
- Switch Interface

3. Media Interfaces 170

- Ethernet, Utopia, ADASL, ...

4. GPIO Preprocessor 126

- Preprocessing functions

5. OPB Master/Slave Function 134

- Customer Proprietary Function
- USB 1.1->
- Ethernet MAC, GPIO, UART, RS232, etc.

5 In addition to the above functions, the FPGA can provide customized preprocessing functions to accelerate table lookup actions, for example, for frame classification or frame forwarding. External coprocessors or Content Addressable Memories (CAMs) often require a unique interface protocol, along with external interposer chips that may be eliminated with internal FPGA functions. These may include:

-Construct a table lookup key from predetermined sub fields within the packet header or payload

-Preformat an information packet for transfer to an external coprocessor or CAM

-Hash functions to reduce the length of a lookup key

-Append control information to message frames passed to external coprocessors

-Remove control information from message frames received from external coprocessors

-State-machine to handle handshaking protocol with an external coprocessor or CAM

-Pre filter functions to reduce load on external coprocessors

Accordingly, common logic from network processor to network processor is implemented utilizing standard cells while the high risk or customizable logic is implemented utilizing FPGA cells. It should be understood that although particular functions have been identified as ones to implement utilizing FPGA cells, one of ordinary skill in the art recognizes that a variety of functions could be provided to be customized. In addition, one recognizes that

although certain functions within the processor 100 were chosen to be implemented as FPGA cells and certain functions were chosen to be implemented as standard cells, experience and/or empirical evidence may indicate that some of the functions may become common logic that are now determined to be high risk and vice versa.

5 Using the above-described Field-Programmable Network Processor, a business method can be utilized to advantageously produce such customizable network processors. In order to mass-produce chips for the broadest possible market, some techniques must be available for each customer to provide unique functions, interfaces, etc. Through the use of a Field Programmable Network Processor (FPNP) for the production of very high volume chips the need for uniqueness is met, without the need for software development and its attendant performance limitations. The FPGA portion could utilize only the on chip FPGA, but may also include off chip FPGA support logic (or a combination of the two).

Typically a customer would have the ability to use the FPGA for unique functions, interfaces, etc. It would also be possible to program, using standard FPGA techniques the chips in the field to upgrade existing hardware, or in the final stages of manufacturing to customize the hardware just prior to shipment. If, as the specific application matures (and the volumes warrant it), then a lower cost custom chip could be produced for that customer.

For a description of this feature, refer now to the following discussion in conjunction with the accompanying figures. Figure 2 is a block diagram of a system for customizing a field programmable network processor (FPNP 254). A customer provides logic files 202 and a verification module to a vendor 250. The vendor includes a FPGA programmer, A FPNP and a FPNP verification bench 256. Figure 3 is a flow chart, which illustrates a method for customizing a field programmable network processor in accordance with the present invention.

Referring to Figures 2 and 3 together, a customer initially wants to customize the FPNP 254 with an application-specific function. After designing and verifying his logic in software or in a stand-alone FPGA chip, a logic file 202 which includes a logic description along with a verification module 204 sent to the FPNP vendor 250 for inclusion on the FPNP 254 via step 302. The FPNP vendor can customize the FPNP chip to the customer's specification using standard FPGA programming techniques, via step 304, and verify the resulting network processor using a combination of the FPNP verification bench 256 and the verification vectors from the verification module 204 supplied by the customer 200, via step 306. The FPNP vendor 250 can then mass-program and verify chips for the end customer. The customer 200 does not have to maintain a verification environment for the entire FPNP complex, but only for his unique logic. The customer 200 also does not have to maintain the amount of FPGA programming systems needed to mass-produce the part.

A field programmable network processor in accordance with the present invention combines the flexibility and time-to-market of a general purpose network process with the ability to customize portions of the silicon in the field. This can be accomplished by adding field-programmable gate array (FPGA) cells to the general purpose network processor chip. Customer-specific functions, high-risk logic and I/O interfaces can be implemented in the FPGA portion of the chip in the field using standard FPGA programming techniques. FPGA cells can be manufactured on the same silicon die as standard cell gates using existing manufacturing techniques.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the

[illegible]